# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/652,109 | 08/29/2003 | David L. Kaminsky | RSW9-2003-0112US1 (107) | 9838 |

46320      7590      12/11/2007

CAREY, RODRIGUEZ, GREENBERG & PAUL, LLP
STEVEN M. GREENBERG
950 PENINSULA CORPORATE CIRCLE
SUITE 3020
BOCA RATON, FL 33487

| EXAMINER |
|---|
| TERMANINI, SAMIR |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2178 | |

| MAIL DATE | DELIVERY MODE |
|---|---|
| 12/11/2007 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**MAILED**

**DEC 1 1 2007**

**Technology Center 2100**

## BEFORE THE BOARD OF PATENT APPEALS
## AND INTERFERENCES

Application Number: 10/652,109
Filing Date: August 29, 2003
Appellant(s): KAMINSKY ET AL.

Scott D. Paul
For Appellant

**EXAMINER'S ANSWER**

This is in response to the appeal brief filed 8/30/2007 appealing from the Office action mailed 5/31/2007.

**(1) Real Party in Interest**

A statement identifying by name the real party in interest is contained in the brief.

**(2) Related Appeals and Interferences**

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

**(3) Status of Claims**

The statement of the status of claims contained in the brief is correct.

**(4) Status of Amendments After Final**

No amendment after final has been filed.

**(5) Summary of Claimed Subject Matter**

The summary of claimed subject matter contained in the brief is correct.

**(6) Grounds of Rejection to be Reviewed on Appeal**

The Appellants' statement of the grounds of rejection to be reviewed on appeal is correct.

**(7) Claims Appendix**

The copy of the appealed claims contained in the Appendix to the brief is correct.

## (8) Evidence Relied Upon

6,601,233          Underwood          7-2003

## (9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

### CLAIM REJECTIONS-35 U.S.C. §101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 1-5 are rejected under 35 U.S.C. §101 because the claimed invention is directed to non-statutory subject matter, and further raises questions as to whether the claims are directed to an abstract idea. As an initial matter, claims 1–5 lack the necessary physical articles or objects to constitute a machine or a manufacture within the meaning of 35 U.S.C. 101. Just as they are clearly not a series of steps or acts, to be a process, nor are they a combination of chemical compounds to be a composition of matter. Claims 1-5, fail to fall within a statutory category.

More specifically, Claims 1-5 are directed to computer programs claimed as computer listings *per se*, i.e., the descriptions or expressions of software programs because the "System" limitation recited, in claims 1-5, is

not limited to a hardware system. In like manner, the term "system" is wholly unsupported by any physical steps or structure because the claims fail to positively recite any concrete, physical implementation of the "system."

Under the broadest reasonable interpretation of the claims, consistent with the Specification, Appellants' software-only "system" does not define any structural and functional interrelationship between the software and any elements of a computer which permit the software system's functionality to be realized.

## CLAIM REJECTIONS-35 U.S.C. §102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. §102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 1-13 are rejected under 35 U.S.C. 102(e) as being anticipated by *Underwood* (US Pat. No. 6,601,233 B1).

With respect to independent **claim 1**, *Underwood* teaches a system (e.g., "...a hardware implementation...," col. 11, lines 9-10) for autonomically configuring widgets (e.g. "...a collection of largely autonomous components, called objects...." col. 11, lines 44-55)(emphasis added). The objects are "...user interface objects [which] may include one or more of the following: a push button, a text box, a text area, a radio button, a check box, a drop down, a blank item, a user interface list, and a static table. (col. 62, lines 46-57). The user can interact by, "clicking on one of the user interface objects, changing text in one of the interface objects, exiting a text box of one of the interface objects. Further, the user action involving one of the user interface objects may cause a predetermined event." (col. 62, lines 47-57).

*Underwood's* discloses various examples of widgets programmed to be disposed in a user interface using framework services. More particularly, a User Interface Framework is taught to program widgets for disposal within a HTML user interface using components (i.e. "The User Interface framework provides components for generating HTML. An [sic] HTML page is generated from a combination of the various UI Components." col. 63, lines 1-5; see also "FIG. 20A illustrates a method 2000 for generating a graphical user interface." col. 62, lines 34-35). The User Interface Framework implements these services through, inter alia, the "AFForm" COM object. To illustrate, below is the table showing the component (i.e., AFForm) for programming widgets to be disposed in a HTML user interface:

| Component | Generates |
| --- | --- |
| AFForm | Form containing the widgets |
| AFPushButton | Push button widget |
| AFTextBox | Single-line entry text box widget |
| AFTextArea | Multi-line entry text box widget |
| AFRadioButton | Radio button widget |
| AFCheckBox | Check box widget |
| AFDropDown | Combo box widget |
| AFBlankItem | Blank item widget (used for spacing) |
| AFUIList | Single-Select List Box widget - IE4 Only |
| AFStaticTable | Static Table widget |
| AFHardCodedASPAction | Javascript function - Move to next page |
| AFJScriptAction | HTML - attach Javascript function to a form element |
| AFScriptGenerator | Javascript tag and functions |
| AFStyleSheet | Cascading style sheet (CSS) |

(col. 63, lines 13-29). In an embodiment, Underwood explains that the AFForm component is used in conjunction with form element widgets to build the user interfaces,

> "Initially, the application creates an instance of the form component and sets its attributes. Following this activity, the application creates instances of the associated form element widgets and <u>adds them to the form using the form's add method.</u>"

(col. 63, lines 53-64) (emphasis added). In particular, this is the "add method" referred to by *Underwood*:

| Int Add(Widget Object, eventcollection) | Add a widget object to this form. Widgets are created separately. |
| --- | --- |
| String generate(event-collection | Generates the HTML code for the Form. The return value is the output HTML and should be printed to the screen. |

(col. 63, lines 24-27).

*Underwood* teaches at least one widget comprising a dynamically configurable presentation field, (e.g. list box or text boxes) as disclosed in this passage:

> "The AFUIList component creates a sophisticated DHTML based single-select list box form widget. The list box widget consists of a fixed headings row and a scrollable set of data rows. <u>The list box widget supports data entry through data row level associated check boxes and text boxes</u>.

(col. 68, lines 40-53) (emphasis added).

In like fashion, the presentation field taught in *Underwood* is dynamically configurable. For example, "The AFUIList view captures the values and updates the state of the list box to reflect the user choice." (col. 68, lines 54-56). Moreover, the presentation field can be dynamically configured via the setValue and getValue methods which are taught to set and get values of the business component instance of a drop down field:

> "The AFViewDropDownBOMapping component defines the mapping between a user interface drop down field and the business component instances containing the value to display. This class gets/sets an UI field value by getting/setting the business component instance contained in the activity context....This interface provides the setValue and getValue methods used to set and get values of the business component instance."

(col. 40, lines 27-38). The disclosure in *Underwood* is not only comprehensive in its disclosure of dynamically configurable list boxes, but also as to various types of User Interface widgets, shown below:

| | |
|---|---|
| AFViewDynamicBOMapping<br>AFVBViewDynamicBOMapping | Defines the mapping between a dynamically created user interface entry field and the business component instances containing the value to display. |
| AFViewTextAreaBOMapping<br>AFVBViewTextAreaBOMapping | Defines the mapping between a user interface multi-line entry field and the business component instances containing the value to display. |
| AFViewDropDownBOMapping<br>AFVBViewDropDownBOMapping | Defines the mapping between a user interface drop down combo box field and the business component instances containing the value to display. |
| AFViewUIListBOMapping<br>AFVBViewUIListBOMapping | Defines the mapping between a user interface Selected List Box field and the business components containing the values to display. |

(col. 31, lines 33-48).

*Underwood* demonstrates, in one embodiment for instance, business rules for validating input through an input prompt by dynamically configuring a recommended set of options which can be activated in the widget:

> "[using] externally stored parameters and validation rules [f]or example, an application may be designed to retrieve the tax rate for the State of Illinois. When the user enters "Illinois" on the screen, the application first validates the user's entry by checking for its existence on the "State Tax Table", and then retrieves the tax rate for Illinois. Note that codes tables provide an additional degree of flexibility. If the tax rates changes, the data simply needs to be updated; no application logic needs to be modified."

(col. 119, lines 3-12).

*Underwood* also discloses that, "...a plurality of business components in a business are first defined with each logical business component having a plurality of capabilities." (col. 2, lines 6-9). A "policy" comprising a plurality of business components is taught to be embedded into the business components ("For example, embedding too much policy information can lead to a Business

Component that is more difficult to maintain and customize.," col. 321, lines 35-38)(emphasis added).

*Underwood* makes clear that the widgets are further configured according to their context (e.g. activity context):

> "GetValueForUIField Return the value for the UI field mapped to an instance of a business component contained in the activity context. If the business component instance is not part of the activity, then return the default value for the UI field."

(col. 37, lines 10-18). The "setCodesTable" rules engine is configured to process the business rules with a call, "Int setCodesTable(String): Populate dropdown box with a Codes Table value" (col. 68, lines 10-13)

During prosecution, Appellants raised the question as to how the limitations in the claims, correspond to features in the cited prior art. Therefore, pursuant to MPEP §1207.02 (A)(9)(e), the following table is provided:

| Language of Claim 1. | Quotation from prior art. | Specific citation. |
|---|---|---|
| A system for autonomically configuring a user interface comprising: | "...a hardware implementation..." | Column 11, lines 9-10 |
| at least one widget programmed to be disposed in | "The form includes a plurality of attribute rules dictating a manner in which user interface objects are situated thereon." | Column 62, lines 36-39 |
| | "The user interface objects may include one or more of the following: a push button, a text | Column 62, lines 46-57 |

| | | |
|---|---|---|
| | box, a text area, a radio button, a check box, a drop down, a blank item, a user interface list, and a static table." | |
| the user interface, | "FIG. 20A illustrates a method 2000 for generating a graphical user interface." | Column 62, lines 34-35 |
| | "The User Interface framework provides components for generating HTML. An [sic] HTML page is generated from a combination of the various UI Components." | Column 63, lines 1-5 |
| said at least one widget comprising a dynamically configurable presentation field; | "The AFUIList component creates a sophisticated DHTML based single-select list box form widget. The list box widget consists of a fixed headings row and a scrollable set of data rows. The list box widget supports data entry through data row level associated check boxes and text boxes. | Column 68, lines 40-53 |
| | "The AFViewDropDownBOMapping component defines the mapping between a user interface drop down field and the business component instances containing the value to display. This class gets/sets an UI field value by getting/setting the business component instance contained in the activity context....This interface provides the setValue and getValue methods used to set and get values of the business component instance..." | Column 40, lines 27-38 |
| and, a policy | "For example, embedding too much policy information can lead to a Business Component | Column 321, lines 35-38 |

| | that is more difficult to maintain and customize." | |
|---|---|---|
| comprising a plurality of business rules | "...a plurality of business components in a business are first defined with each logical business component having a plurality of capabilities." | Column 2, lines 6-9 |
| for configuring said at least one widget in the user interface | "...externally stored parameters and validation rules. For example, an application may be designed to retrieve the tax rate for the State of Illinois. When the user enters "Illinois" on the screen, the application first validates the user's entry by checking for its existence on the "State Tax Table", and then retrieves the tax rate for Illinois. Note that codes tables provide an additional degree of flexibility. If the tax rates changes, the data simply needs to be updated; no application logic needs to be modified." | Column 119, lines 3-12 |
| based upon a context provided by said at least one widget; and, | "GetValueForUIField    Return the value for the UI field mapped to an instance of a business component contained in the activity context. If the business component instance is not part of the activity, then return the default value for the UI field." | Column 37, lines 10-18). |
| a rules engine configured to process said business rules. | "setCodesTable(String): Populate dropdown box with a Codes Table value" | Column 68, lines 10-13 |

As to dependent **claim 2**, *Underwood* further teaches that the widget is configured to be disposed in a markup language document ("The User Interface framework provides components for generating HTML. An [sic]

HTML page is generated from a combination of the various UI Components." col. 63, lines 1-5). The preferred embodiment in *Underwood* discloses widgets configured to be disposed in a markup language:

> "A preferred embodiment of the invention utilizes HyperText Markup Language (HTML) to implement documents on the Internet together with a general-purpose secure communication protocol for a transport medium between the client and a company."

(col. 15, line 61 -to- col. 16, line 8).

As to dependent **claim 3**, *Underwood* further teaches that that business rules specify at least one suggested option to be presented to an end user through said at least one widget (e.g. "...get the Customer Object from the Activity Context and add the default values...," col. 282, lines. 35-36; see also e.g. "[p]opulate dropdown box with a Codes Table value," col. 68, lines 10-13).

As to dependent **claim 4**, *Underwood* teaches business rules specifying at least one option which is not to be presented to an end user through said at least one widget ("...externally stored parameters and validation rules, [for w]hen the user enters "Illinois" on the screen, the application first validates the user's entry by checking for its existence on the 'State Tax Table'...," col. 119, lines 3-12).

As to dependent **claim 5**, *Underwood* further teaches the business rules specifying rules for validating input provided through the presentation field ("...utilize externally stored parameters and validation rules. For

example, an application may be designed to retrieve the tax rate for the State of Illinois. When the user enters "Illinois" on the screen, the application first validates the user's entry by checking for its existence on the "State Tax Table", and then retrieves the tax rate for Illinois." col. 119, lines 3-12).

As to independent **claim 6**, *Underwood* teaches a method for autonomically configuring a user interface widget "a plurality of attribute rules dictating a manner in which user interface objects are situated thereon.," col. 62, lines 36-39; See also, "The user interface objects may include one or more of the following: a push button, a text box, a text area, a radio button, a check box, a drop down, a blank item, a user interface list, and a static table." col. 62, lines 46-57), comprising the steps of: evaluating business rules for configuring the user interface widget according to context information for the user interface widget ("GetValueForUIField Return the value for the UI field mapped to an instance of a business component contained in the activity context. If the business component instance is not part of the activity, then return the default value for the UI field.," col. 40, lines 27-38); and, configuring the user interface widget with options permitted by said evaluation. (""The AFViewDropDownBOMapping component defines the mapping between a user interface drop down field and the business component instances containing the value to display. This class gets/sets an UI field value by getting/setting the business component instance contained in the activity context....This interface provides the setValue and

getValue methods used to set and get values of the business component instance...," col. 40, lines 27-38).

During prosecution, Appellants raised the question as to how the limitations in the claims correspond to features in the cited prior art. Therefore, pursuant to MPEP §1207.02 (A)(9)(e), the following table is provided:

| Language of Claim 6 | Quotation from prior art. | Specific citation. |
|---|---|---|
| a method for autonomically configuring a user interface widget comprising the steps of: | "a plurality of attribute rules dictating a manner in which user interface objects are situated thereon." | Column 62, lines 36-39 |
| | "The user interface objects may include one or more of the following: a push button, a text box, a text area, a radio button, a check box, a drop down, a blank item, a user interface list, and a static table." | Column 62, lines 46-57 |
| evaluating business rules for configuring the user interface widget | "The AFViewDropDownBOMapping component defines the mapping between a user interface drop down field and the business component instances containing the value to display. This class gets/sets an UI field value by getting/setting the business component instance contained in the activity context....This interface provides the setValue and getValue methods used to set and get values of the business | Column 40, lines 27-38 |

| | component instance..." | |
|---|---|---|
| according to context information for the user interface widget; | "GetValueForUIField Return the value for the UI field mapped to an instance of a business component contained in the activity context. If the business component instance is not part of the activity, then return the default value for the UI field." | Column 40, lines 27-38 |
| and, configuring the user interface widget with options permitted by said evaluation. | "...externally stored parameters and validation rules. For example, an application may be designed to retrieve the tax rate for the State of Illinois. When the user enters "Illinois" on the screen, the application first validates the user's entry by checking for its existence on the "State Tax Table", and then retrieves the tax rate for Illinois. Note that codes tables | Column 37, lines 10-18).<br><br><br><br><br><br><br><br>Column 119, lines 3-12 |

As to dependent **claim 7**, *Underwood* further teaches the configuring step comprising the step of suggesting at least one option to be presented to an end user through said user interface widget (e.g., "The AFUIList view captures the values and updates the state of the list box to reflect the user choice.," col. 68, lines 54-56; see also e.g. "[p]opulate dropdown box with a Codes Table value," col. 68, lines 10-13).

As to dependent **claim 8**, *Underwood* further teaches the configuring step comprising the step of filtering at least one option from being presented to an end user through said user interface widget ("...externally stored parameters and validation rules, [for w]hen the user enters "Illinois" on the screen, the application first validates the user's entry by checking for its existence on the 'State Tax Table'...," col. 119, lines 3-12).

As to dependent **claim 9**, *Underwood* further teaches the configuring step comprising the step of validating input provided through a presentation field in said user interface ("...utilize externally stored parameters and validation rules. For example, an application may be designed to retrieve the tax rate for the State of Illinois. When the user enters "Illinois" on the screen, the application first validates the user's entry by checking for its existence on the "State Tax Table", and then retrieves the tax rate for Illinois. " col. 119, lines 3-12).

As to dependent **claims 10-13**, these claims differ from claims 6-9, respectively, only in they are directed to products defined by the processes of claims 6-9, respectively. Since *Underwood* teaches the autonomic configuring of the user interface widgets by "A computer program embodied on a computer readable medium" (col. 326, lines. 41-42)-these claims are rejected for the same reasons set forth in the treatment of claims 6-9, respectively.

**(10) Response to Argument**

I.  THE REJECTION OF CLAIMS 1-5 UNDER
     35 U.S.C. § 101

Appellants argue (Reply Brief dated 8/30/2007 at p. 4.) that:

> The Examiner has neither cited any case law that supports the Examiner's position nor set forth a cogent argument as to why a system does not necessarily refer to a hardware system. In this regard, Appellants note that claim 1 recites "at least one widget programmed to be ... ". Software alone is incapable of being "programmed." Instead, software alone is the program. Thus, the claimed at least one widget cannot be software alone, as alleged by the Examiner.

The Examiner respectfully disagrees. Assuming, *arguendo*, that the word "system" in the claim's preamble is to be construed as structurally limiting. While it may be true, that any structurally limiting terminology in the preamble must be treated as a claim limitation – to do so in this case, would ignore the "light" of the Specification (*Corning Glass Works v. Sumitomo Elec.* U.S.A., Inc., 868 F.2d 1251, 1257, 9 USPQ2d 1962, 1966 (Fed. Cir. 1989). By way of example, Appellants' Specification contains only two figures of which, "Figure 1 is <u>a block diagram illustrating a system</u> for autonomically configuring user interface widgets in a user interface..." (Specification, p. 6)(Emphasis added)(Note: The remaining Figure 2 depicts a process flow chart for autonomically configuring a user interface widget in the system of Figure 1).

For clarity, the "system" illustrated by Figure 1 of Appellants' Specification, is reproduced below:

FIG. 1
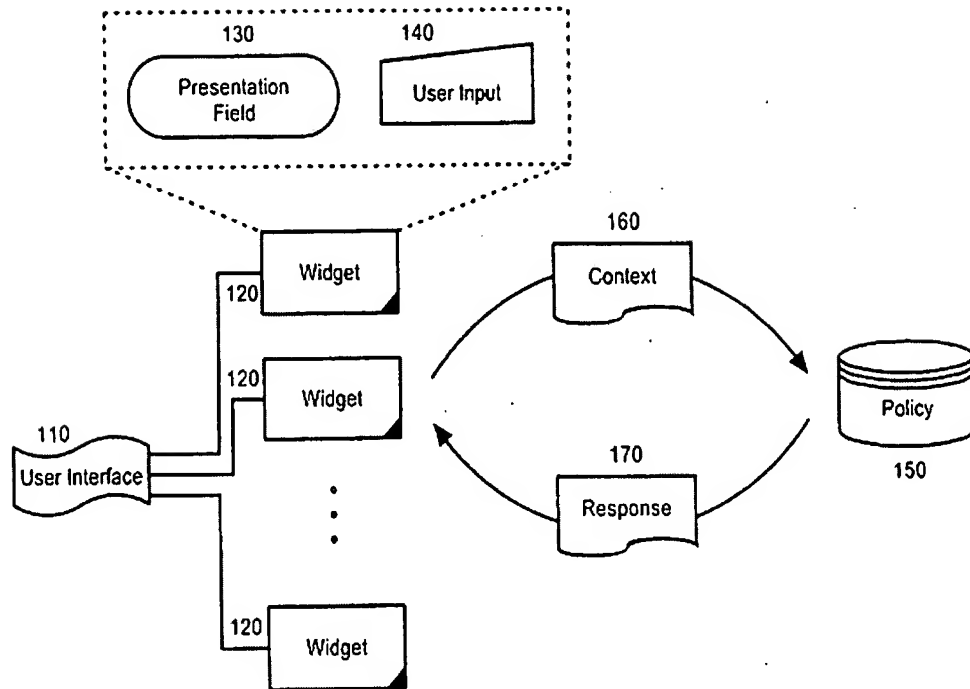
As can been seen from figure 1, the system has been illustrated without any hardware and instead had been illustrated exclusively by software blocks. Equally important, Appellants explicate that "[t]he present invention can be realized in hardware, <u>software</u>, <u>or</u> a combination of hardware and software." (Emphasis added)(Specification, second paragraph, page 11). In other words, the "system" recited in claims 1-5, read in light of the Specification and accompanying Figures, leaves open the possibility the word "system" would be reasonably interpreted by one of ordinary skill in the art to be a collection of software-only components, excluding those that are "computer-readable."

Appellants argue (Reply Brief dated 8/30/2007 at p. 5.) that:

> The Examiner appears to have a fundamental misunderstanding as to what defines the claimed invention. The Examiner's citation to page 11 of Appellants' specification is not dispositive as to what defines the claimed invention because the claims set forth the metes and bounds of Appellants' claimed invention, not the specification.

The Examiner respectfully disagrees. Notwithstanding exemplary embodiments in the Specification, Appellants have not provided lexicographic definitions for the following claimed limitations: a system for autonomically configuration a widget; a dynamically configurable presentation field; a policy; a business rule; a context; and a rules engine. Additionally, Appellants have not claimed their invention using language falling under the scope of 35 U.S.C. 112, 6th paragraph. Therefore, for at least these two reasons, the Examiner further submits that the scope of the claimed subject matter is defined by the broadest reasonable interpretation that is consistent with the Specification.

Furthermore, As can been seen from figure 1, the system has been illustrated without any hardware and instead had been illustrated exclusively by software blocks. In the same way, Appellants spell out that "[t]he present invention can be realized in hardware, software, or a combination of hardware and software." (Specification, second paragraph, page 11). Therefore, the "system" recited in claims 1-5, read in light of the Specification and accompanying Figures, leaves open the possibility the word "system" would be reasonably interpreted by one of ordinary skill in the art to

be a collection of software-only components, excluding those that are "computer-readable."

For at least these reasons, the Examiner respectfully submits that Appellants have not advanced a reason for discarding the broadest reasonable interpretation of the claims, consistent with the Specification, in favor of an interpretation that may limit the claims in such a way to require statutory subject matter. It is during patent examination, in which the claims <u>must be</u> interpreted as broadly as their terms reasonably allow. See *In re American Academy of Science Tech Center*, 367 F.3d 1359, 1369, 70 USPQ2d 1827, 1834 (Fed. Cir. 2004). In other words, the pending claims <u>must</u> be "given their broadest reasonable interpretation consistent with the specification" (*Phillips v. AWH Corp.*, 415 F.3d 1303, 75 USPQ2d 1321 (Fed. Cir. 2005) and with the interpretation that those skilled in the art would reach. (*In re Cortright*, 165 F.3d 1353, 1359, 49 USPQ2d 1464, 1468 (Fed. Cir. 1999).

Appellants argue (Reply Brief dated 8/30/2007 at p. 5.) that:

Moreover, the Examiner's implied assertion (see also the first full paragraph on page 4 of the Second Office Action) that claims, given their broadest interpretation, can be both directed to statutory subject matter (e.g., hardware) and non-statutory subject matter (i.e., software alone) are deemed non-statutory is both legally unsupported and has considerable consequences apparently not considered by the Examiner.

The Examiner respectfully disagrees. Appellants have argued against an assertion that the Examiner did not make. Appellants conceded this fact when they started their, above-quoted argument, with "...the Examiner's implied assertion..." (emphasis added). The Examiner made no such assertion implied or express. Conversely, the Examiner asserted that it is during patent examination, in which the claims must be interpreted as broadly as their terms reasonably allow. See *In re American Academy of Science Tech Center*, 367 F.3d 1359, 1369, 70 USPQ2d 1827, 1834 (Fed. Cir. 2004). Appellants' arguments exploring the consequences of claims having boundaries, uncertain in scope, are addressed by 35 U.S.C. § 112, second paragraph – and not 35 U.S.C. §101.

Therefore, For at least these reasons, the Examiner respectfully submits that the "system" recited in claims 1-5, read in light of the Specification and accompanying Figures, leaves open the possibility the word "system" would be reasonably interpreted by one of ordinary skill in the art to be a collection of software-only components.

Appellants argue (Reply Brief dated 8/30/2007 at p. 6-7.) that:

Instead, 35 U.S.C. § 101 only requires that the claimed invention cover statutory subject matter and does not explicitly prevent a patent from issuing if the claimed invention also covers non- statutory subject matter.

The Examiner respectfully disagrees. Appellants have confused the meaning of 35 U.S.C. § 101 as it applies to the issue at hand. The issue is not:

"[Does] 35 U.S.C. § 101...explicitly prevent a patent from issuing if the claimed invention also covers non-statutory subject matter[?]"

Instead, it is:

Interpreted as broadly as the term reasonably allows, does the "system" recited in claims 1-5, in light of the Specification and Figures, leave open the probability that it would be reasonably interpreted by one of ordinary skill in the art to be only a collection of software-only components?

If this question is answered in the affirmative, Appellants' arguments exploring what 35 U.S.C. § 101 does not explicitly prevent, need not reached, for it is during patent examination, in which the claims must be interpreted as broadly as their terms reasonably allow. See *In re American Academy of Science Tech Center*, 367 F.3d 1359, 1369, 70 USPQ2d 1827, 1834 (Fed. Cir. 2004).

Appellants argue (Reply Brief dated 8/30/2007 at p. 6-8.) that:

> Appellants are claiming a system for autonomically configuring a user interface. Moreover, the useful, concrete, and tangible result (i.e., autonomically configuring a user interface) is clearly identified in the preamble of claim 1. In this regard, the Examiner has failed to explain why the system, encompassed by the limitations recited in claim 1, fails to produce this useful, concrete, and tangible result.

The Examiner respectfully disagrees. As discussed above, the claimed "system" is a description or expression of a software system. Computer programs claimed as computer listings *per se*, (i.e., the descriptions or expressions of the programs), are not physical "things." They are <u>neither</u> computer components nor statutory processes, as they are not "acts" being performed. Such claimed <u>computer programs</u> do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer, which permit the computer program's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a computer program is a computer element which defines structural and functional interrelationships between the computer program and the rest of the computer which permit the computer program's functionality to be realized, and is thus statutory. See *Lowry*, 32 F.3d at 1583-84, 32 USPQ2d at 1035.

II.    <u>THE REJECTION OF CLAIMS 1-13 UNDER 35 U.S.C. § 102 FOR ANTICIPATION BASED UPON UNDERWOOD</u>

Appellants argue (Reply Brief dated 8/30/2007 at p. 8.) that:

As argued by Appellants on page 4 of the First Response, "the Examiner's citations to column 125 and column 316 are <u>completely silent</u> with regard to 'configuring said at least one widget in the user interface,' as claimed" (emphasis in original). As such, these cited passages are irrelevant to the claimed limitation.

In addition, on p. 11 argue:

As argued by Appellants on page 4 of the First Response, "the Examiner's citations to column 125 and column 316 are completely silent with regard to 'configuring said at least one widget in the user interface,' as claimed" (emphasis in original). As such, these cited passages are irrelevant to the claimed limitation.

The Examiner respectfully disagrees. Appellants have not provided lexicographic definitions for "...configuring said at least one widget in the user interface...." Additionally, Appellants have not claimed their invention using language falling under the scope of 35 U.S.C. 112, 6th paragraph. The aforementioned language does not preclude the following teachings: "FIG. 20A illustrates a method 2000 for generating a graphical user interface." col. 62, lines 34-35; and "The User Interface framework provides components for generating HTML. An [sic] HTML page is generated from a combination of the various UI Components." col. 63, lines 1-5. Appellant has not argued why any of the teachings fail to teach the claimed limitations. The scope of the recited "...configuring said at least one widget in the user interface...." encompasses an extremely broad range of configuring.

Appellants argue (Reply Brief dated 8/30/2007 at p. 12.) that:

The Examiner is not permitted to parse limitations to such a fine degree as to eliminate the meaning of the limitation. Otherwise, nearly all claims could be anticipated by a dictionary or reference book in the general art. For example, if a claim was to a "green dog," and a reference showed both a blue dog and a green tree, one cannot properly parse "green" from "dog" so as to have the blue dog identically disclose "dog" and the green tree identically disclose "green." This parsing, however, is exactly what was performed by the Examiner. The above reproduced passages do not have any apparent relationship to one another, and the Examiner is not permitted to mix and match unrelated teachings solely to arrive at the claimed limitations without any guidance from the teachings of the applied prior art.

The Examiner respectfully disagrees. In their argument, Appellants are mix and matching unrelated things. The Examiner warns that a dictionary, by its very nature, is something entirely different from what was cited as evidence of anticipation. Using the idiom for a false analogy, Appellants are "comparing apples and oranges." All else being equal, dictionaries are directed to the words of a language -- not the topic of Appellants' invention. Dictionaries teach the meaning of words-*Underwood* teaches Appellants invention. Appellants hypothetical dictionary probably does not have the same Abstract found in *Underwood's*, to wit:

A method of generating software based on business components. A plurality of logical business components in a business are first defined with each business component having a plurality of capabilities. Next, functional interrelationships are identified between the logical business components. Code modules are then generated to carry out the capabilities of the logical business components and the functional interrelationships between the logical business components, wherein the code modules represent a transformation of the logical business components to their physical implementation, while ensuring the capabilities that are carried out by each code module are essentially unique to the logical business component associated with the code module. Next, the functional aspects of the code modules and the functional relationships of the code modules are tested. The code modules are then subsequently deployed in an e-commerce environment.

Due to the volume of the cited reference, Appellants' prefer to only consider the locations cited by the examiner *in vacuum*, that is, without the context and defined meanings of the cited teachings by other portions of the same reference. Appellant should note that "[t]he use of patents as references [a]re part of the literature of the art, relevant for all they contain." (*In re Heck*, 699 F.2d 1331, 1332-33, 216 USPQ 1038, 1039 (Fed. Cir. 1983) (quoting *In re Lemelson*, 397 F.2d 1006, 1009, 158 USPQ 275, 277 (CCPA 1968)). A reference may be relied upon for all that it would have reasonably suggested to one having ordinary skill the art, including non-preferred embodiments. *Merck & Co. v. Biocraft Laboratories*, 874 F.2d 804, 10 USPQ2d 1843 (Fed. Cir.), cert. denied, 493 U.S. 975 (1989). This is true even if teachings do not materialize in proximity to other teachings within the same reference. To hold otherwise would preclude the use of comprehensive references.

Appellants argue (Reply Brief dated 8/30/2007 at p. 12.) that:

> The Examiner reliance on In re Heck and Merck & Co. is misplaced. The fact that references may be relied upon for all that they teach does not abrogate, from the Examiner, the burden of setting forth a proper rejection. (emphasis in original).

The Examiner respectfully disagrees. Setting forth a proper rejection does not require the examiner to cite portions of a reference that are only close in proximity, to each other. It appears that the appellant is proposing that the teachings are not related to each other because they do not materialize in proximity to other teachings within the same reference. Again, a reference may be relied upon for all that it would have reasonably suggested to one having ordinary skill the art, including non–preferred embodiments. *Merck & Co. v. Biocraft Laboratories*, 874 F.2d 804, 10 USPQ2d 1843 (Fed. Cir.), cert. denied, 493 U.S. 975 (1989). This is true even if teachings do not materialize in proximity to other teachings within the same reference. To hold otherwise would preclude the use of comprehensive references.

*Underwood*, in its entirety, is directed towards Appellants invention, and all of the teachings cited by the examiner, as discussed above, read on Appellants' invention.

The Examiner's analysis should not be an invitation to Appellants to engage and guessing and/or speculation as to how the Examiner is interpreting the elements of the claims, what specific features within the applied prior art the Examiner believes identically disclose the claimed invention, and the basis for the Examiner's interpretation and belief. Moreover, the Examiner's analysis should not be an invitation to Appellants to comb through a reference for "the context and defined meanings of the cited teachings." The cited reference of Underwood is not the typical 20-30 page reference. On the contrary, Underwood is 278 pages long, with 111 sheets of drawings, and about 2 columns worth of claims with the rest being written (approximately 164 pages or 328 columns). By advocating that Appellants review Underwood to make up for the deficiencies in the Examiner's own analysis, the Examiner has clearly failed to meet the initial burden of establishing a prima facie case of anticipation

The Examiner respectfully disagrees. Appellants were not asked to "comb through a reference for 'the context and defined meanings of the cited teachings.'" The meaning of each of the cited portions heretofore, were clear and recognizable to one of ordinary skill in the pertinent art.

Appellants have invented in this Field and it is respectfully submitted that the Examiner's responsibilities under 37 CFR §104(c)(2) are satisfied with properly set forth *prima facie* rejections. The task of tutoring the Appellants, on subject matter of the prior art reference, is unduly burdensome, particularly when identity of terminology is present, and one of ordinary skill in the pertinent art would recognize the cited portions read on Appellants' claimed invention.

Appellants argue (Reply Brief dated 8/30/2007 at p. 15.) that:

As apparent from this passage, it appears that Underwood consider "business objects" to be a particular discrete portion used in a business computer application, and the relationship the "business objects" have with the claimed "policy comprising a plurality of business rules" is also unclear to Appellants.

The Examiner points out that Appellants are not addressing the Examiners rejections, as applied, to each of the claimed limitations. Moreover, the "business objects" and "policy" recited in the claims, are not limited the way appellant suggests.

Appellants argue (Reply Brief dated 8/30/2007 at p. 16.) that:

> underlined passage is identically disclosed. Column 32 states adding a user interface component to "the UI context of the activity," but this is not the same as using context provided by a widget configure the widget. Also, the cited passage in column 280 states that "[t]he views map the UI widgets to attributes of business objects," and this passage is also silent as to business rules that configure a widget based upon a context provided by the widget

The Examiner points out that that the words: "...UI context..." is, inter alia, the widget's context. Furthermore, the "context" and "configuring" recited in the claims, are not limited the way appellant imply.

Appellants argue (Reply Brief dated 8/30/2007 at p. 16.) that:

> Activity Context and add the default values." How this passage is (i) related to "business rules," (ii) teaches that at least one suggested option is specified by the business rules; and (iii) teaches that the option is presented through the widget is completely unclear to Appellants.

Appellants argue (Reply Brief dated 8/30/2007 at p. 17.) that:

The Examiner's analysis, however, is completely silent as to relating these teachings back to the claimed "business rules" (i.e., the Examiner's cited passages in columns 125 and 315) and to the claimed "configuring said at least one widget in the user interface based upon a context provided by said at least one widget" (i.e., the Examiner cited passage in column 32

The Examiner points out that the "relationship back" to the claimed "configuring," inter alia, is taught by a, "business component contained in the activity context." (col. 37, lines 10-18). Furthermore, the argued meaning of said "business rules" is not limited the way Appellants imply.

### (11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

X _Samir Jermanini_

Samir Termanini
Patent Examiner Art Unit 2178

STEPHEN HONG
SUPERVISORY PATENT EXAMINER

Conferees:

X _____

Stephen Hong
Supervisory Patent Examiner, GAU 2178

X _____

Doug Hutton,
Supervisory Patent Examiner GAU 2176